

OPERADOR PARA ESCONDER E RENOMEAR EVENTOS

Alexandre Mota & Augusto Sampaio

Hiding

- Operador para construir processos:

$$P \setminus X$$

dado um processo P e um conjunto de eventos X

- $P \setminus X$ comporta-se como P exceto pelo fato de que os eventos em X
 - ▣ não podem ser observados, são escondidos
 - ▣ são executados internamente, de forma automática

- Útil para esconder eventos e comunicações auxiliares do sistema:
 - ▣ concentrar entendimento nos aspectos essenciais do sistema, esquecer os detalhes internos da implementação
 - ▣ evitar comunicações não desejadas ou não permitidas
 - ▣ garantir comunicação ponto-a-ponto:
$$(P \mid X \mid Q) \setminus X$$
$$(P \mid X \mid Y \mid Q) \setminus (X \cap Y)$$

Exemplo

$\text{COPY}'(a, b) = a?x \rightarrow b!x \rightarrow \text{COPY}'(a, b)$

$\text{BCHAIN}(N) =$

$|| \ i : \{0..N-1\} \ @ \ [\{ | \bar{d}\bar{d}.i, \bar{d}\bar{d}.i+1 | \}]$
 $\text{COPY}'(\bar{d}\bar{d}.i, \bar{d}\bar{d}.i+1)$

$\text{HCHAIN}(N) =$

$\text{BCHAIN}(N) \setminus \{ | \bar{d}\bar{d}.i \ | \ i \leftarrow \{1..N-1\} | \}$

Para funcionar como um buffer!

Exemplo

$\text{BN}(a, b, N, s) =$

$\#s < N \ \& \ a?x \rightarrow$

$\text{BN}(a, b, N, s^{<x>})$

$[] \ \#s > 0 \ \& \ b!head(s) \rightarrow$

$\text{BN}(a, b, N, tail(s))$

$\text{BN}(dd.0, dd.5, 5, <>) [= \text{HCHAIN}(5)$

$\text{HCHAIN}(6) [= \text{BN}(dd.0, dd.6, 6, <>)$

Equivalente? E BCHAIN?

$$(P \mid \sim \mid Q) \setminus X = (P \setminus X) \mid \sim \mid (Q \setminus X)$$

$$(P \setminus Y) \setminus X = P \setminus (X \cup Y)$$

$$(P \setminus Y) \setminus X = (P \setminus X) \setminus Y$$

$$P \setminus \{ \} = P$$

$$\begin{aligned} (a \rightarrow P) \setminus X &= P \setminus X && (\text{se } a \in X) \\ &= a \rightarrow (P \setminus X) && (\text{se } a \notin X) \end{aligned}$$

Exercícios

$$(a \rightarrow P) \setminus \{a\} = ?$$

$$(a \rightarrow P) \setminus \{b\} = ?$$

$$(a \rightarrow P \text{ [] } b \rightarrow Q) \setminus \{a, b\} = ?$$

$$(a \rightarrow P \text{ [] } b \rightarrow Q) \setminus \{b\} = ?$$

Respostas dos exercícios

$$(a \rightarrow P) \setminus \{a\} = P \setminus \{a\}$$

$$(a \rightarrow P) \setminus \{b\} = a \rightarrow P \setminus \{b\}$$

$$(a \rightarrow P \ [\] \ b \rightarrow Q) \setminus \{a, b\} \\ = P \setminus \{a, b\} \ |\sim| \ Q \setminus \{a, b\}$$

$$(a \rightarrow P \ [\] \ b \rightarrow Q) \setminus \{b\} \\ = (a \rightarrow P \setminus \{b\} \ [\] \ Q \setminus \{b\}) \\ |\sim| \ Q \setminus \{b\}$$

Processos com e sem restrição de visibilidade

```
(c?x:A -> P) \ X
= if X ∩ A = {} then
    c?x:A -> (P \ X)
else
    (
        (c?x:(A \ X) -> (P \ X))
        [ ] ( | ~ | { P(a) \ X | a ∈ X ∩ A } )
    )
| ~ | ( | ~ | { P(a) \ X | a ∈ X ∩ A } )
```

Exemplo

```
ATM2 =
  incard?c -> pin.fpin(c) -> req?n ->
    ((dispense!n -> outcard.c -> ATM2)
     |~| (refuse ->
          (ATM2 |~| outcard.c -> ATM2)))

ATMref(n) =
  (n<Max) & refill -> ATMref(Max)
[] incard?c -> pin.fpin(c) -> req?w ->
  (if w<=n then
    dispense!w -> outcard.c ->
    ATMref(n-w)
  else refuse -> outcard.c -> ATMref(n)
)
```

Exemplo

? $ATM2 \sqsubseteq ATMref(0)$

? $ATM2 \sqsubseteq ATMref(0) \setminus \{refill\}$

? $ATMref(0) \setminus \{refill\} \sqsubseteq ATM2$

? $ATMref(0) \sqsubseteq ATM2$

$ATMref(0)$ é determinística

$ATMref(0) \setminus \{refill\}$ é não-determinística

Divergência

- Processo realiza seqüência infinita de eventos internos
- O operador de *hiding* pode introduzir divergência ou *livelock*:

$$P = (\mu X. a \rightarrow X) \setminus \{a\}$$

$$Q = a \rightarrow Q \setminus \{a\}$$

$$P = P \quad (P = (\mu X. X))$$

$$P = \text{div}$$

Recursão guardada x construtiva

$$(P \mid [X] \mid Q) \setminus Z \\ = (P \setminus Z) \mid [X] \mid (Q \setminus Z)$$

(**se** $X \cap Z = \{\}$)

$$(P \mid [X \mid Y] \mid Q) \setminus Z \\ = (P \setminus Z \cap X) \mid [X \mid Y] \mid (Q \setminus Z \cap Y)$$

(**se** $X \cap Y \cap Z = \{\}$)

Traces

- $\text{traces}(P \setminus X) =$
 $\{ s \setminus X \mid s \in \text{traces}(P) \}$
- $s \setminus X$ denota $s \uparrow (\text{Events} \setminus X)$

Exercícios

- Do livro texto
 - ▣ Essenciais: 3.1.1
 - ▣ Opcionais: 3.1.2

OPERADOR PARA RENOMEAR EVENTOS

Alexandre Mota & Augusto Sampaio

Renomeação

- Operador para construir processos:

$P[[R]]$

dado um processo P e uma relação entre eventos R na forma

$e_1 \leftarrow f_1, e_2 \leftarrow f_2, \dots, e_n \leftarrow f_n$

- R determina a renomeação dos eventos de P :
 - e_i em P é mapeado para f_i em $P[[R]]$
- Útil para criar vários processos similares

Renomeação

- Caso R seja uma função sobre os eventos de P ,
 - ▣ $P[[R]]$ é obtido a partir da sintaxe de P pela renomeação dos eventos segundo R
 - ▣ se R for injetiva, $P[[R]]$ oferece o evento $R(a)$ exatamente quando P oferece a
 - ▣ caso contrário, o comportamento do processo renomeado pode mudar :

```
(  a  -> P
  [] b  -> Q)[[a <- b, b <- b]]
```

Exemplo

$\text{COPY} = \text{left?x} \rightarrow \text{right!x} \rightarrow \text{COPY}$

$\text{COPY}'(a,b) = a?\text{x} \rightarrow b!\text{x} \rightarrow \text{COPY}'(a,b)$

$\text{COPY}[[\text{left} \leftarrow \text{aa}, \text{right} \leftarrow \text{bb}]]$
 $[= \text{COPY}'(\text{aa}, \text{bb})]$

$\text{COPY}'(\text{aa}, \text{bb})$
 $[= \text{COPY}[[\text{left} \leftarrow \text{aa}, \text{right} \leftarrow \text{bb}]]]$

Exemplo

```
SPLIT = in?x -> if x%2==1 then
                out1.x -> SPLIT
                else out2.x -> SPLIT
```

```
RenSPLIT =
    SPLIT[[in.x <- in', out1.x <- out1',
           out2.x <- out2' | x <- T]]
```

```
SPLIT' = in' -> (    out1' -> SPLIT'
                  |~| out2' -> SPLIT')
```

```
RenSPLIT = SPLIT'
```

Renomeação

- Caso R não seja uma função,
 - ▣ $P[[R]]$ oferece um evento em $R(a)$ exatamente quando P oferece o evento a , caso o domínio de R contenha todos os eventos de P
 - ▣ pode-se introduzir escolhas que não existiam antes:

$$\begin{aligned} & (a \rightarrow \text{STOP})[[a \leftarrow a, a \leftarrow b]] \\ = & (a \rightarrow \text{STOP}) [] (b \rightarrow \text{STOP}) \end{aligned}$$

Leis

$$(P \mid \sim \mid Q)[[R]] = P[[R]] \mid \sim \mid Q[[R]]$$

$$(P [] Q)[[R]] = P[[R]] [] Q[[R]]$$

$$(\text{?x:A} \rightarrow P)[[R]] =$$

$$\text{?y:R(A)} \rightarrow$$

$$\mid \sim \mid \{P(z)[[R]] \mid z \in A \wedge zRy\}$$

Quando R é funcional, Renomeação distribui sobre mais operadores, como as hiding e (variações de) paralelismo

Traces

- $\text{traces}(P[[R]]) =$
 $\{ t \mid \exists s \in \text{traces}(P). s R^* t \}$
- $s R^* t \text{ sse}$
 - ▣ $\#s = \#t$
 - ▣ $\forall i < \#s + 1. s(i) R t(i)$

Exercícios

- Do livro texto
 - ▣ Essenciais: 3.2.3